CAMPUS
DE EXCELENCIA
INTERNACIONAL

Graduado en Ingeniería Informática

Universidad Politécnica de Madrid

Facultad de Informática

TRABAJO FIN DE GRADO

Transformación de series temporales numéricas a secuencias simbólicas

Autor: ENRIQUE NICOLÁS SANZ

Director: JUAN PEDRO CARAÇA VALENTE Y HERNÁNDEZ

MADRID, JUNIO DE 2013

DEPARTAMENTO DE LENGUAJES, SISTEMAS INFORMÁTICOS E INGENIERÍA DE SOFTWARE

FACULTAD DE INFORMÁTICA. UNIVERSIDAD POLITÉCNICA DE MADRID

TRANSFORMACIÓN DE SERIES NUMÉRICAS TEMPORALES EN SECUENCIAS SIMBÓLICAS

Autor

Enrique Nicolás Sanz

Director

Juan Pedro Caraça-Valente Hernández

Doctor en Informática

Junio, 2013

ÍNDICE GENERAL

RESUMEN

El Trabajo de Fin de Grado aborda el tema del Descubrimiento de Conocimiento en series numéricas temporales, abordando el análisis de las mismas desde el punto de vista de la semántica de las series.

La gran mayoría de trabajos realizados hasta la fecha en el campo del análisis de series temporales proponen el análisis numérico de los valores de la serie, lo que permite obtener buenos resultados pero no ofrece la posibilidad de formular las conclusiones de forma que se puedan justificar e interpretar los resultados obtenidos.

Por ello, en este trabajo se pretende crear una aplicación que permita realizar el análisis de las series temporales desde un punto de vista cualitativo, en contraposición al tradicional método cuantitativo. De esta forma, quedarán recogidos todos los elementos relevantes de la serie temporal que puedan servir de estudio en un futuro.

Para abordar el objetivo propuesto se plantea un mecanismo para extraer de la serie temporal la información que resulta de interés para su análisis. Para poder hacerlo, primero se formaliza el conjunto de comportamientos relevantes del dominio, que serán los símbolos a mostrar en la salida de la aplicación. Así, el método que se ha diseñado e implementado transformará una serie temporal numérica en una secuencia simbólica que recoge toda la semántica de la serie temporal de partida y resulta más intuitiva y fácil de interpretar.

Una vez que se dispone de un mecanismo para transformar las series numéricas en secuencias simbólicas, se pueden plantear todas las tareas de análisis sobre dichas secuencias de símbolos. En este trabajo, aunque no se entra en este post-análisis de estas series, sí se plantean distintos campos en los que se puede avanzar en el futuro. Por ejemplo, se podría hacer una medida de la similitud entre dos secuencias simbólicas como punto de partida para la tarea de comparación o la creación de modelos de referencia para análisis posteriores de las series temporales.

ABSTRACT

This Final-year Project deals with the topic of Knowledge Discovery in numerical time series, addressing time series analysis from the viewpoint of the semantics of the series.

Most of the research conducted to date in the field of time series analysis recommends analysing the values of the series numerically. This provides good results but prevents the conclusions from being formulated to allow justification and interpretation of the results.

Thus, the purpose of this project is to create an application that allows the analysis of time series, from a qualitative point of view rather than a quantitative one. This way, all the relevant elements of the time series will be gathered for future studies.

The design of a mechanism to extract the information that is of interest from the time series is the first step towards achieving the proposed objective. To do this, all the key behaviours in the domain are set, which will be the symbols shown in the output. The designed and implemented method transforms a numerical time series into a symbolic sequence that takes in all the semantics of the original time series and is more intuitive and easier to interpret.

Once a mechanism for transforming the numerical series into symbolic sequences is created, the symbolic sequences are ready for analysis. Although this project does not cover a post-analysis of these series, it proposes different fields in which research can be done in the future. For instance, comparing two different sequences to measure the similarities between them, or the creation of reference models for further analysis of time series.

CAPÍTULO 1: INTRODUCCIÓN

El análisis de colecciones de datos ordenados en el tiempo, denominadas series temporales, es fundamental en muchos campos como la ingeniería, la medicina o los negocios.

Estudiar cómo se ha comportado una variable hasta el momento puede ser de gran interés a la hora de predecir su comportamiento futuro. Del mismo modo, determinar qué otras variables han tenido un comportamiento similar puede ayudar a decidir las acciones a tomar, bien sea para conservar la evolución actual o bien para modificarla completamente. Por este motivo, cada vez hay una necesidad mayor de buscar series temporales de datos similares a una serie dada en una base de datos.

Además de la obtención de patrones en series temporales, existen otras muchas tareas en relación con el análisis de series temporales como, por ejemplo, conocer el grado de similitud entre dos series temporales o conocer los comportamientos concretos de una serie temporal que tengan un significado especial en el dominio en cuestión (pendientes acentuadas, pasos por cero, máximos globales., etc.). Las técnicas de descubrimiento de conocimiento y minería de datos aplicadas a series temporales se convierten en herramientas muy útiles para este tipo de tareas.

Centrando la atención en el problema de conocer los comportamientos concretos de la serie temporal que tengan un significado especial en el dominio bajo estudio, parece lógico pensar que una buena solución al problema consiste en identificar las secciones o regiones de la serie temporal donde aparecen dichos comportamientos y dotarlas de contenido semántico en función del significado que tengan en la serie. Llegados a este punto, sería deseable analizar la serie temporal por el contenido semántico de estas secciones en vez de analizar el conjunto de los valores numéricos que tome la serie temporal en cada instante.

1.1. Objetivos

En el presente trabajo se pretende crear una aplicación cuya interfaz de definición de símbolos permita realizar una especificación de comportamientos que ocurren en las series temporales de un dominio, donde cada comportamiento se identificará con un símbolo, que estará dotado de una serie de atributos.

Así mismo, se diseñará e implementará un método que sea capaz de transformar la serie numérica en otra simbólica, identificando las características más relevantes de las diferentes secciones de la serie temporal numérica, lo que permitirá extraer el conocimiento relevante contenido en esta serie.

La finalidad de este proceso es la reducción de la dimensionalidad de los datos (es decir, poder trabajar con un menor número de datos), pudiendo centrar el foco en la información más relevante, y poder aplicar determinadas técnicas para futuros estudios, como la Programación Genética Dirigida por Gramáticas.

1.2. Estructura del Trabajo

El presente trabajo se ha estructurado en varios capítulos cuyo contenido se presenta a continuación:

- *Capítulo 1*. Introducción: Ya discutido en esta memoria, en él se ha hecho un análisis de la importancia de obtener la información relevante de las series numéricas temporales, haciendo un acercamiento a los campos en los que es más necesario. También se han explicado los objetivos buscados en el trabajo, detallando lo que se va a necesitar para construir la aplicación que nos permita llevarlo a cabo.

- *Capítulo 2*. Entorno del Trabajo: En este capítulo se hace un análisis de las áreas que son relevantes para este trabajo. Primeramente se hace una introducción sobre cómo se debe hacer una buena recolección de información en distintos

campos. Después, se concreta este análisis en el que atañe al trabajo, las series temporales y los distintos métodos existentes para recolectar la información semántica de las mismas.

- *Capítulo 3*. Desarrollo del Problema: Este capítulo contiene toda la información referente al desarrollo de la aplicación, desde la información con la que se partía al inicio del problema, detallando la especificación de requisitos, hasta la propia resolución del mismo, profundizando en el diseño y diversos detalles asociados a él. El capítulo finaliza con un análisis de los resultados del sistema, que son los datos que realmente nos interesan y servirán para futuros estudios.

- *Capítulo 4*. Conclusiones: El capítulo incluye las reflexiones y conclusiones a las que se ha llegado tras el trabajo realizado.

- *Capítulo 5*. Futuras líneas de trabajo: Este capítulo recoge las posibles líneas de investigación futuras que surgen de la investigación realizada.

- *Capítulo 6*. Bibliografía: Este capítulo recoge todas las fuentes de información que se han tomado como referencia para la realización del trabajo.

- *Anexo.* En el último apartado se recoge un anexo con el Diagrama de Clases de la aplicación.

CAPÍTULO 2: ENTORNO DE TRABAJO

En la actualidad, el exceso de información debido a la era digital provoca una sobrecarga de datos. Por ese motivo, es cada vez más importante mejorar los modelos utilizados para poder analizar y entender toda la información disponible, además de mejorar las técnicas utilizadas para obtener y almacenar los datos. Este es el marco de investigación donde se encuadra este capítulo, en el que se describen las técnicas y algoritmos que existen en la actualidad en el contexto del análisis de datos y en particular en el análisis de series temporales.

El presente capítulo está dividido en dos secciones. En la primera sección de describe el proceso de descubrimiento de conocimiento y en la segunda se explican las técnicas de transformación de series temporales en secuencias simbólicas. Ambas secciones toman la mayoría de información, aunque adaptada para los intereses de este trabajo, de **[Santamaría et al. 11]**, ya que la investigación que ahí se lleva a cabo toma muchos puntos de partida similares a este.

2.1. Descubrimiento de Conocimiento

El descubrimiento de conocimiento en bases de datos (**Knowledge Discovery in Databases - KDD**) permite encontrar información útil en una colección de datos. El conocimiento se puede representar de muchas formas, siendo una de ellas, la utilización de reglas que permiten describir las propiedades de esos datos. La obtención de patrones o clases de objetos de los datos es el objetivo que persigue el proceso KDD. Actualmente, la tecnología nos permite obtener y almacenar cada vez más datos de una forma sencilla, pero el análisis de esos datos es lento y caro debido al crecimiento exponencial de los datos. El incremento de datos en una base de datos puede deberse al aumento del número de registros (N), individuos u objetos, contenidos en la misma, y también al número de campos (d), o atributos, de cada objeto. A medida que crece el volumen de datos, la manipulación de estos y su exploración se hace más difícil, sobre todo utilizando las capacidades humanas.

Las personas buscan la tecnología para poder automatizar la búsqueda de conocimiento en los datos. En la extracción de conocimiento a partir de una gran base de datos se suelen utilizar multitud de técnicas: matemáticas, estadísticas, de reconocimiento de patrones, de inteligencia artificial, de sistemas expertos, de visualización de datos, de búsqueda y de razonamiento.

Los campos donde el proceso KDD puede ser aplicado son muy variados, incluyendo datos médicos, aplicaciones financieras, datos científicos, etc. Con este proceso se podría por ejemplo detectar el fraude en el área de la telecomunicación, conocer la asociación de ventas en el área de los supermercados u obtener una segmentación de los clientes de un banco.

KDD es un proceso iterativo e intuitivamente interactivo y no un sistema que automáticamente analiza los datos dando como resultado conocimiento útil. Aunque se disponga de una gran cantidad de datos encima de la mesa, no se debe esperar sacar conclusiones útiles a primera vista. El usuario de un sistema KDD debe tener un conocimiento sólido del dominio del problema; de este modo, el propio usuario será capaz inicialmente de separar la información en una serie de subconjuntos. Este proceso está compuesto por una serie de pasos o etapas que son las que se describen a continuación:

1. Entender perfectamente el dominio de aplicación y los objetivos que se buscan, que es donde se encuentra el conjunto de datos.

2. Seleccionar un conjunto de datos (datos objetivo) que serán los que se utilicen para aplicar los pasos que se citan a continuación, tantas veces como sea necesario, para la búsqueda de conocimiento.

3. Limpieza y preprocesamiento de los datos: incluye operaciones básicas como la eliminación de ruido o datos no apropiados, la decisión de qué atributos de los datos se van a utilizar y el tratamiento de los atributos con valores no recogidos.

4. Transformación de los datos: en este paso se deben buscar las características útiles e importantes que caracterizan a los datos. Además, se debe reducir el número de atributos o encontrar una representación invariante para los dato

mediante métodos de transformación o métodos de reducción de la dimensionalidad.

5. Descubrimiento de patrones mediante la utilización de las técnicas de minería de datos (data mining). En esta fase se deben elegir los algoritmos de data mining que se consideren oportunos para la búsqueda de patrones en los datos.

6. Postprocesamiento o Interpretación: en este paso se tienen que interpretar los patrones que se han descubierto. Se eliminan los patrones redundantes o irrelevantes, trasladando la información útil a términos inteligibles por el usuario.

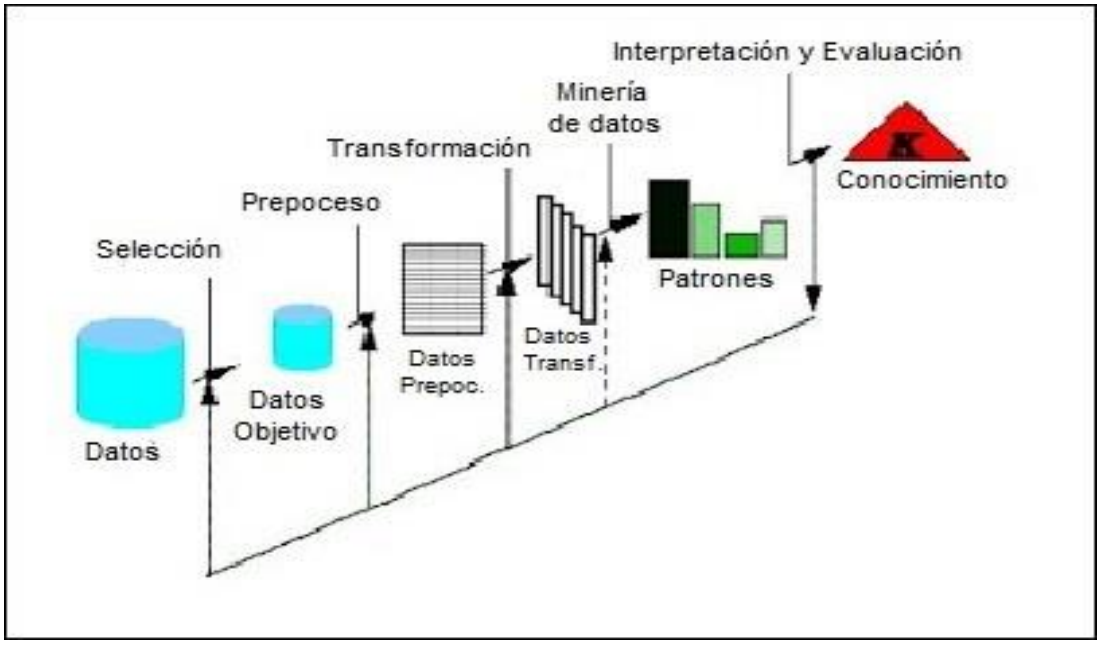


Figura 1. Pasos del proceso KDD [3]

2.2. Análisis y Transformación de Series Temporales

Como ya se ha comentado, el análisis de colecciones de datos ordenados en el tiempo, las series temporales, es fundamental en campos como el deporte, la medicina, la economía o la ingeniería. Estudiar el comportamiento de distintas variables hasta un

momento concreto puede ser muy útil para predecir su comportamiento en adelante. Igualmente, determinar qué otros valores han tenido un comportamiento similar puede ayudar a decidir las acciones a tomar, bien sea para conservar la evolución actual o bien para modificarla radicalmente.

Poder comparar series de datos temporales similares a las de bases de datos, no simplemente haciendo rastreos secuenciales, sino encontrando métodos o técnicas que ayuden a disminuir dichos rastreos, se está convirtiendo en algo cada vez más necesario para las empresas. Teniendo en cuenta la cantidad de información que hay en bases de datos y el gran tamaño que pueden alcanzar las series temporales, los análisis de éstas tienen una importancia vital en muchos dominios.

Una vez llegados a este punto, el siguiente paso sería definir medidas de distancia para conocer las series temporales que se parecen entre sí o buscar patrones de comportamiento dentro de una serie temporal. Así, después podrían aplicarse técnicas de Minería de Datos para extraer el conocimiento significativo, como se explica en el proceso KDD, en la sección 2.1.

Dado que este trabajo no pretende abarcar todo este proceso KDD, en este apartado nos centraremos en analizar las distintos técnicas de transformación que se pueden aplicar a los datos una vez que estos han sido preprocesados, que es la parte que nos concierne dado el módulo software que se pretende diseñar.

Para que se pueda realizar una medida de similitud entre series temporales, se necesita un previo paso de limpieza y transformación de las series temporales. En muchos casos, dependiendo del método de transformación seleccionado, la representación de las secuencias afecta al análisis de Minería de Datos que se hará más adelante.

En nuestro caso, nos centraremos únicamente en las transformaciones de series numéricas temporales en secuencias de caracteres, sin abordar las que hacen transformaciones en secuencias numéricas similares.

En la figura se presentan los distintos tipos de transformaciones que exploraremos: **SDL, SDA y SAX**, en las que se basa en gran parte el módulo software a crear.

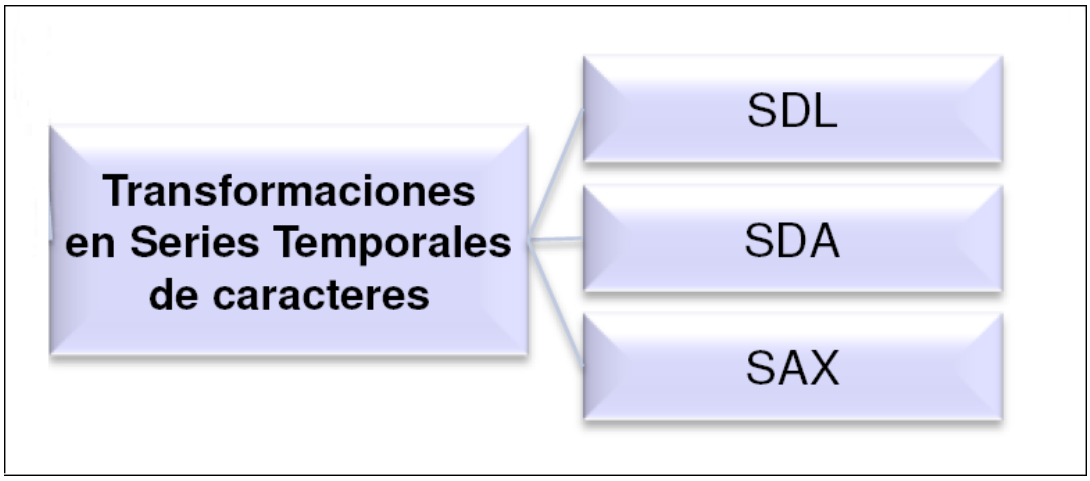


Figura 2. Transformaciones de Series Numéricas Temporales

Como se ha mencionado, el objetivo de este tipo de transformación es convertir una serie temporal, compuesta por números, en una serie compuesta por palabras o símbolos. De este modo se dispondrá de una serie de símbolos que caracterizarán a la secuencia temporal a la cual representan. Estos símbolos suelen ser dependientes del dominio de la aplicación que se esté considerando.

2.2.1. Shape Definition Language (SDL)

En primer lugar, tratamos el lenguaje de definición de formas (**SDL – Shape Definition Language**) [Sang-Wook et al. 06], para la obtención de objetos basados en formas contenidas en históricos asociados con estos objetos. Cada objeto de la base de datos tiene asociado un nombre que explica el comportamiento de la serie temporal en un intervalo concreto. A los valores de ese intervalo se les denomina historia. De ese modo, cada elemento de la historia es un símbolo que identifica el comportamiento de la serie temporal en una unidad de tiempo (subida, estable, bajada suave,...). Se trata de un pequeño pero potente lenguaje que permite una amplia variedad de consultas sobre las formas encontradas en los históricos temporales.

La sintaxis del lenguaje sería la siguiente:

Lenguaje (símbolo lb ub (iv) (fv))

Expresión 1. Definición del lenguaje SDL

El primer término representa un símbolo del lenguaje; *lb* y *ub* son descriptores qu representan el límite inferior (*lower bound*) y superior (upper *bound*) de la variación permitida desde el valor inicial al valor final de la transición. Los últimos dos, *iv* y *fv* , son opcionales y especifican restricciones de valor inicial (*initial value*) y final (*final value*) de la transición. A continuación se muestra un ejemplo de símbolos y valores prefijado para sus descriptores:

Símbolo	Descripción	lb	ub
up	aumento ligero de la transición	0.05	0.19
Up	aumento grande de la transición	0.20	1.0
down	disminución ligera de la transición	-0.19	-0.05
Down	disminución grande de la transición	-1.0	-.19
Stable	el valor final casi igual al valor inicial	-0.04	0.04
Zero	tanto el valor inicial como el final son cero	0	0

Figura 3. Tabla del lenguaje SDL

Teniendo en cuenta este ejemplo, aplicamos la transformación a símbolos de la siguiente serie temporal:

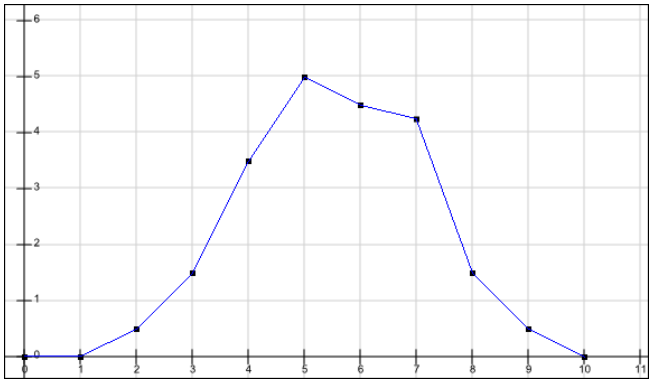


Figura 4. Serie Temporal para SDL

Y nos quedaría la siguiente expresión:

(zero stable up up up down stable Down down stable)

Expresión 2. Transformación de la serie temporal con SDL

Cabe destacar que es en este lenguaje en el que se basa en mayor medida el módulo software que se va a diseñar, ya que cubre de forma bastante coherente, aunque algo incompleta, la información que queremos mostrar de las series temporales a analizar.

2.2.2. Shape Description Alphabet (SDA)

A continuación, se presenta el alfabeto de descripción de formas (**SDA – Shape Definition Alphabet**) para codificar la forma de la serie temporal en un alfabeto de caracteres, de forma que pueda ser tratada como texto y utilizar esta transformación para una estructura de indexación de una dimensión. Como se podrá intuir, se trata de un **subconjunto del alfabeto de SDL**, pero suficiente para describir cualquier serie a analizar.

Primero se traduce la secuencia en un texto de secuencia de caracteres. Seguidamente se desplaza una ventana por el texto, a partir de la cual se va creando la firma y se va mapeando el string en un número de bits de la firma. La firma se almacena en el fichero de firmas con un puntero al bloque de texto (representación 1D de la secuencia) que ha sido utilizado para crear la firma. Para lanzar consultas, simplemente se calcula la firma de la secuencia consulta 1D y se hace una búsqueda lineal en el fichero de firmas. Las ventajas de este método respecto al anterior son las siguientes:

- El fichero de firmas es pequeño.
- La simplicidad de probar hace que la búsqueda sea bastante rápida. Además tiene una complejidad O(n) en el tiempo.

Este método es suficiente para describir cualquier serie temporal. En la siguiente tabla se muestra un ejemplo con cinco símbolos:

Símbolo	Descripción	lvalue	hvalue
a	aumento grande de la transición	5	-
u	aumento ligero de la transición	.2	4.99
s	transición estable	-1.99	1.99
d	disminución ligera de la transición	-4.99	-2
e	disminución grande de la transición	-	-5

Figura 5. Tabla del lenguaje SDA

Se calcula el valor diferencia entre dos puntos adyacentes y, dependiendo del intervalo del símbolo, definido por *Ivalue* y *hvalue*, se asignará un símbolo u otro. Estos valores son dependientes del dominio y es necesario encontrar los valores óptimos para el dominio que se pretende analizar.

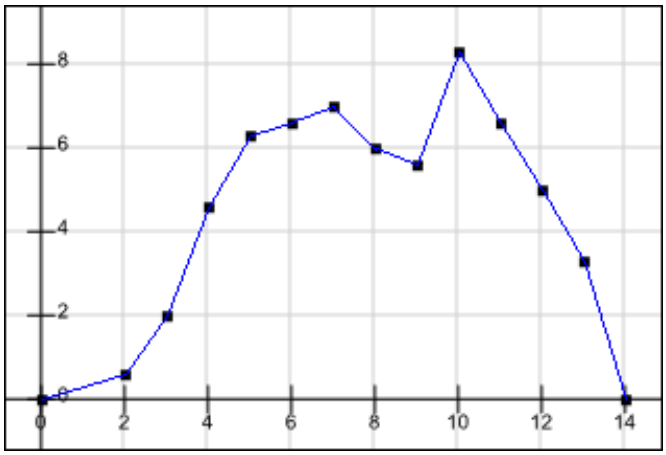


Figura 6. Serie temporal para SDA

Así, en este caso, nos quedaría esta expresión:

(u u a u s s d s a e e e e)

Expresión 3. Transformación de la serie temporal con SDA

Como se puede intuir, obteniendo esta salida y sin conocer exactamente la serie de entrada, podrían obtenerse series distintas al realizar la traducción inversa, ya que sólo se nos indica el comportamiento de los tramos, pero no sobre los puntos exactos en

donde ocurren. Al igual que el anterior método, también se basará en parte el módulo a crear para hacer la transformación de símbolos.

2.2.3. Symbolic Aggregate Approximation (SAX)

Por último, se presenta una aproximación para transformar series temporales en secuencias simbólicas (**SAX – Symbolic Aggregate ApproXimation**) con dos objetivos:

- Reducir la dimensionalidad de la serie temporal.
- Cumplir que la distancia entre dos series temporales sea menor o igual que la distancia de las dos series temporales originales para, de esta forma, poder utilizar técnica de Minería de Datos.

La aproximación consta de dos partes:

La primera transforma la serie temporal C = c ,..., cn mediante PAA (***Piece Aggregate Approximation***) para conseguir reducir la dimensionalidad de la serie, pasando de tener *m* elementos representarse por un espacio w-dimensional por un vector $\overline{C} = \bar{c}_1,\ldots,\bar{c}_w$.

En la figura 7 se muestra esta primera parte:

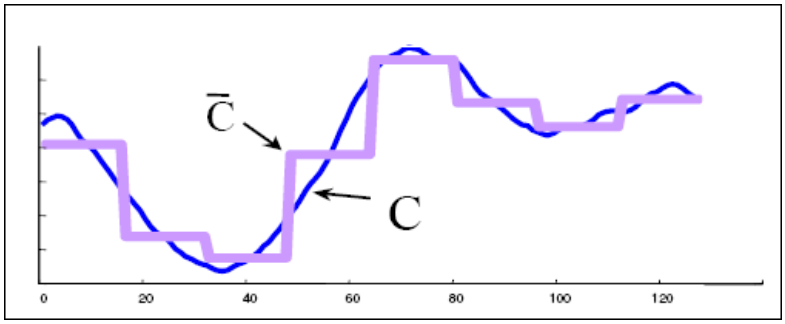


Figura 7. Primera parte de SAX. PAA

La segunda parte consiste en convertir en valores discretos mediante símbolos los segmentos obtenidos del PAA. Teniendo en cuenta que las series temporales siguen una distribución Gaussiana, se procede a utilizar esta distribución para obtener los valores

límite que definen las regiones, a partir de los cuales un elemento de $\overline{C}$, dependiendo de la región donde se encuentre, se traducirá al símbolo que representa esa región. En la Figura 8 se muestra un ejemplo donde la longitud de c es 128, la longitud de $\overline{C}$ es 8 y el número de regiones de corte es 3, dando como resultado la secuencia simbólica $\overline{C}$ = b a a b c c b c.

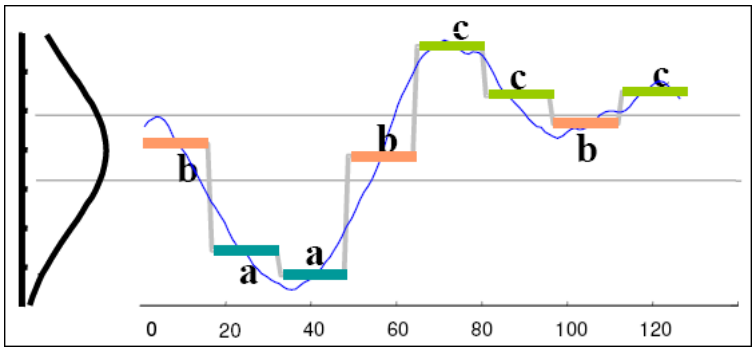


Figura 8. Segunda parte de SAX. Traducción de PAA a símbolos SAX

A diferencia de los dos métodos anteriores, de este no tomaremos información para nuestra aplicación, ya que es distinto el enfoque que se hace para la transformación de la serie. Sin embargo, es importante reseñarlo para saber que, en otras circunstancias, podría ser deseable hacer así las transformaciones en secuencias simbólicas.

CAPÍTULO 3: DESARROLLO DEL PROBLEMA

En el presente capítulo se van a tratar los aspectos que conciernen a la aplicación creada, empezando por el planteamiento inicial que se hizo de la misma, que incluye la motivación que lleva a hacerla y la Especificación de Requisitos Software del sistema.

Después se pasa a hablar del diseño de la aplicación, pasando por diferentes aspectos relacionados que se consideran de interés, como temas matemáticos relativos al análisis de series temporales, añadidos posteriores y las partes que presentaron mayor dificultad.

3.1. Planteamiento del Problema

En esta sección se discuten todos los aspectos previos al diseño e implementación de la aplicación.

3.1.1. Motivación

El trabajo que aquí se presenta se centra en el análisis de los datos de series temporales como medio de obtener unos resultados fácilmente inteligibles por los expertos y conseguir así un alto grado de satisfacción en ellos que propicie un mayor interés por su parte en el uso de herramientas de data mining.

La idea tras la realización de este trabajo es que en muchos dominios, aparte de ser necesario obtener patrones numéricos de comportamiento en series temporales o conocer el valor de similitud entre dos secuencias temporales, también lo es conocer los comportamientos concretos de la serie temporal (picos, subidas agudas, valores constantes, etc.) que tengan un significado especial en el dominio en cuestión. En estos casos, sería deseable poder identificar las secciones o regiones de la serie temporal donde aparecen dichos comportamientos y poder analizar las series temporales por el contenido semántico de estas secciones y no sólo por los valores numéricos que pueda tomar la serie temporal en cada instante. Se podría traducir toda la serie temporal

numérica a otra serie temporal simbólica donde cada uno de los símbolos tuviera un significado relevante en el dominio. Una secuencia temporal simbólica de este tipo aporta un valor añadido en el análisis de la serie al llevar implícito contenido semántico. Este contenido semántico permite caracterizar la propia secuencia temporal y, además, permite la definición de métodos de análisis y comparación de las series temporales que utilicen los mismos conceptos utilizados por el experto, lo que hace posible abordar el problema de una forma similar a como lo hace el experto y mejorar sustancialmente la explicación y justificación de los resultados obtenidos. Así, los datos numéricos serán transformados en datos simbólicos y todos los métodos y técnicas desarrollados se aplicarán sobre estos datos simbólicos.

Por ello, el trabajo se centra en la creación de una aplicación que debe ser capaz de transformar la serie numérica en otra simbólica, identificando las características más relevantes de las diferentes secciones de la serie temporal. Estas características podrán variar de un dominio a otro, pudiendo ser incluso totalmente diferentes.

Por tanto este método proporcionará un mecanismo a partir del cual una serie numérica será convertida en una secuencia simbólica con contenido semántico. Esta secuencia simbólica caracterizará la serie numérica original a través de un conjunto de conceptos que estarán estrechamente ligados a los utilizados por un experto en dicho dominio.

La secuencia simbólica será de utilidad para los usuarios por sí misma, al resaltar los comportamientos más importantes de la serie temporal desde el punto de vista de cada dominio.

3.1.2. Especificación de Requisitos Software

En este apartado se pretende mostrar el funcionamiento del sistema a diseñar y las acciones que permitirá y no permitirá.

Las especificaciones siguen el esquema definido en el **IEEE 830-1998** [1]. Para la protección de datos se tendrá en cuenta el estándar para la seguridad de la información **ISO/IEC 27001** [2].

Así pues, se muestran las principales funciones y requisitos del sistema:

- El sistema podrá ser usado por cualquier usuario, independientemente de sus conocimientos.
- El sistema dispondrá de toda la ayuda necesaria para que puedan entenderse todas las acciones que éste puede realizar.
- El sistema no podrá ser modificado por ningún usuario ajeno a la creación del mismo.
- El sistema podrá tener abierta una instancia del mismo a la vez, para evitar conflictos entre recursos que cambian dinámicamente.
- El sistema debe proteger la información contenida en las entradas que se le especifiquen y no guardar ningún registro de ella después de la utilización del mismo.
- El sistema permitirá que la salida pueda ser configurada por medio de distintos parámetros.
- El sistema tardará menos de 5 segundos en hacer la operación que se le indique.
- El sistema permitirá copiar la información que aparezca en sus distintos campos y ventanas.
- El sistema debe mostrar la información de salida en columnas bien diferenciadas.
- El sistema especificará un formato concreto a seguir para los archivos de entrada que serán procesados.
- El sistema mostrará mensajes de error siempre que se realicen acciones que no cumplan con lo especificado en las ayudas.

- El sistema no podrá ser ejecutado en entornos que no dispongan de la configuración necesaria para aplicaciones de Microsoft.

3.2. Resolución del Problema

En esta sección se explican detalladamente todos los aspectos concernientes al diseño del módulo software. En el primer apartado se tratan los diagramas básicos de diseño que se habían planteado de inicio, hablando primero de la arquitectura software elegida, pasando después a detallar cómo se han estructurado las clases y qué contiene cada una de ellas y por último haciendo una explicación de cómo funcionará el sistema y los elementos que se considerarán.

En los siguientes apartados se habla de los añadidos que se hicieron más adelante en la aplicación, de todas las consideraciones matemáticas relativas a la transformación de las series y por último de todo aquello que produjo más dificultad tanto en el diseño como en la implementación del sistema.

3.2.1. Diseño de la aplicación

Una vez discutida la motivación que nos lleva a crear esta aplicación, pasamos a hablar de todos detalles del diseño del sistema.

Para empezar, se decidió usar el entorno de trabajo de **Visual Studio** y, dentro de él, el lenguaje de programación C#, ya que muchos de los trabajos e investigaciones que se han hecho sobre series temporales y sus transformaciones y post-análisis se han hecho desde este entorno, de forma que hacer su planteamiento sería más sencillo que en otros.

Para avanzar con seguridad en este entorno es necesario conocer cómo funcionan sus clases, métodos y atributos [4]. Por ello, cuando se quiera ejecutar la aplicación, se necesitará disponer de las bibliotecas actualizadas del Framework .NET (versión 4 o superior).

Dado que ya se sabía que se quería hacer un tratamiento de series temporales mediante su transformación a símbolos y posteriores análisis, se planteó de inicio si era preferible recibir las series temporales como un *stream* de datos o de una forma más sencilla, mediante la definición de sus puntos. Finalmente se prefirió la segunda opción, ya que la definición de símbolos es mucho más intuitiva tanto para quien crea la aplicación como para el experto que haga uso de ella.

Para la realización del sistema, se optó por un sistema basado en el patrón de arquitectura **'Modelo Vista Controlador'** en el que se dividieron los distintos módulos en tres partes bien diferenciadas:

- **Vista**: Contiene la interfaz con la que interactuará el usuario. El conjunto de ventanas que formarán parte de la interfaz gráfica serán explicadas en el diagrama de clases.
- **Controlador**: Implementa la interfaz Vista y procesa la gráfica con valores temporales de entrada para producir unos resultados finales.
- **Modelo**: Contiene las definiciones de Gráfica (es decir, la serie temporal de entrada), de Punto y de Tramo por medio de sus clases. Aquí se definirán los símbolos y las reglas de generación de los mismos. También producirá los resultados finales que, pasando por el controlador, llegarán a la interfaz gráfica que verá el usuario final.

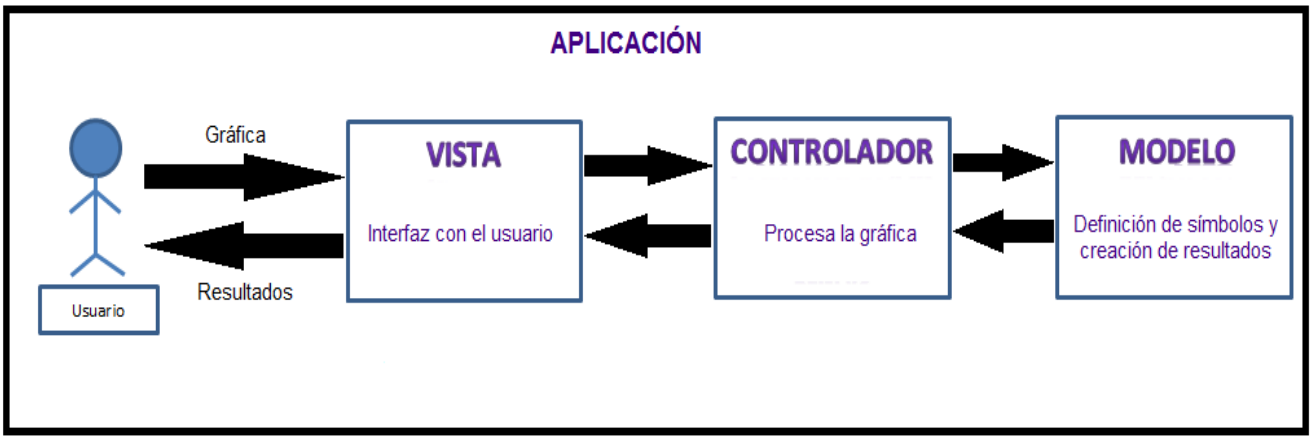


Figura 9. Arquitectura software 'Modelo Vista Controlador'

Así pues, una vez definido cómo iba a estar estructurado el módulo software, se pasó a hacer una definición de los elementos que iba a considerar nuestro sistema.

Por lo investigado en otros trabajos y, tras observar cuáles eran los datos que un experto demanda cuando se dispone a analizar una serie temporal, se decidió incluir los siguientes símbolos como elementos principales del módulo a realizar:

- **Subida:** Símbolo básico que permite saber en qué momento ha aumentado la variable analizada en un periodo de tiempo. Tendrá asociado un atributo que determinará la pendiente que tiene en ese instante, aguda o suave.
- **Bajada:** Al igual que la subida, este símbolo nos dirá cuándo ha disminuido el valor de la variable analizada en un periodo de tiempo. También tendrá al atributo para la pendiente aguda o suave.
- **Constante**: Símbolo que ocurre cuando la variable analizada en la serie temporal se mantiene en el mismo valor durante un periodo de tiempo. Este símbolo no tendrá ningún atributo asociado.
- **Pico**: También llamado máximo, es un símbolo que representa un punto de la serie temporal en que termina de subir un tramo y empieza a bajar. Tendrá asociado el atributo local o absoluto, dependiendo de si ese punto es el de mayor valor de la gráfica (absoluto) o uno del resto (local).
- **Hundimiento**: También llamado mínimo, es lo contrario al pico, un símbolo que ocurre cuando en un punto concreto la serie temporal deja de bajar y empieza a subir. También tendrá el atributo local o absoluto.

Más adelante, como se explica en la sección 3.2.2, se añadió el símbolo **Paso Por Cero**, aunque al principio no se consideró como símbolo por no ser habitual en los análisis de series temporales.

El archivo de entrada con la serie temporal deberá seguir un formato concreto para ser aceptado por la aplicación y procesado y transformado en la secuencia de símbolos. El formato de cada punto debe ser un par de números (debe usarse el carácter 'punto' para los decimales) separados por comas. Para separar un punto de otro, se puede usar el

'punto y coma', el 'salto de línea' o ambos. Los tabuladores no afectan a la lectura del archivo. A continuación se muestra un ejemplo de archivo en el que se consideran todas las opciones posibles:

1,14; 3, 7.45; 5,6; 6,1;
7,0.5
8.56, 23.788

Expresión 4. Ejemplo de serie temporal de entrada

En caso de que el archivo de entrada no cumpla con el formato especificado, se mostraría un mensaje de error.

Dado que la aplicación puede tener algunos elementos que no son intuitivos de comprender, se dispondrá de la **suficiente ayuda** para saber cómo funciona y qué pasos hay que seguir para ejecutar el sistema, por medio de las ventanas pertinentes.

Así pues, para la interacción del usuario, la aplicación contará con una interfaz gráfica cuya ventana principal accederá a las demás ventanas. En ella se podrá abrir un fichero con la serie temporal, acceder a la ayuda o a la configuración de algunos parámetros.

En la barra de estado inferior se mostrará el estado en que se encuentra la ejecución: antes de cargar una serie temporal, antes de procesar la información y después de haber mostrado la salida.

Como parte principal de la aplicación, en la ventana principal habrá un botón con el que se realizará la transformación de símbolos y se mostrarán en un recuadro a su derecha.

Para comprender mejor cómo se ha diseñado todo el sistema, se añade el **Diagrama de Clases** en el Anexo de esta memoria en el que se indican todas las relaciones entre las clases creadas y los atributos y métodos que implementan. Esto incluye la relación que hay entre las ventanas de la interfaz gráfica.

3.2.2. Añadidos posteriores

Aunque el diseño de la aplicación estaba trazado antes de empezar con su implementación, a medida que se avanzó con su creación se decidió incluir una serie de añadidos que no alteraban la planificación original del proyecto.

Para empezar, en el diseño de la aplicación sólo se habían valorado los símbolos subida, bajada, constante, pico y hundimiento (ver sección 3.2.1) y se habían desechado algunos otros como los *puntos de inflexión*, ya que se había estimado que su información no era muy relevante para los resultados del sistema. Sin embargo, una vez que se empezó con la implementación, **se añadió el símbolo de Paso por Cero**, ya que en muchos dominios es relevante saber los momentos en que una variable pasa por el eje de referencia: temperaturas, momentos de relajación de un músculo, alturas, etc.

En segundo lugar, también se había decidido que habría sólo un tipo de ejecución del sistema en el que se mostraría toda la información definida posible sobre la serie temporal de entrada. Sin embargo, dado que eran muchos los contextos en los que podía querer obtenerse información de la serie temporal, se decidió mejorar la forma en que se mostraba la salida, de forma que pudieran obviarse algunos eventos o limitar la aparición de algunos de ellos.

Esto se consiguió por medio de la **adición de tres parámetros** que son configurables por el usuario:

Pendiente límite para subidas/bajadas:

- Este parámetro determina cuándo una subida o bajada deja de ser suave y pasa a ser aguda.
- Si es menor o igual que el valor especificado, será suave; si es mayor, será aguda.
- Por defecto está definido a 1, que se corresponde con una pendiente con ángulo de 45º, tanto para subidas como para bajadas.
- Puede tomar valores decimales desde 0 en adelante.

Altura mínima para picos/hundimientos:

- Este parámetro determina si un pico o hundimiento será considerado como tal en la salida, dependiendo de su altura.
- Si es menor o igual que el valor especificado, no será considerado; si es mayor, sí habrá pico/hundimiento.
- Por defecto está definido a 0, es decir, siempre serán considerados todos los picos y hundimientos.
- Puede tomar valores decimales desde 0 en adelante.
- La altura de un pico/hundimiento es la distancia que hay entre su punto de inicio o fin hasta el punto exacto donde hay pico/hundimiento.

Tiempo en pico/hundimiento:

- Este parámetro determina si un tramo constante, cuyo tramo anterior es una subida y el posterior es una bajada, será considerado como pico/hundimiento o como constante, dependiendo de su duración.
- Si es menor o igual que el valor especificado, habrá pico/hundimiento; si es mayor, será un tramo constante.
- Por defecto está definido a 0, es decir, siempre que haya un tramo constante en las condiciones especificadas, no será considerado pico/hundimiento.
- Puede tomar valores decimales desde 0 en adelante.

El último añadido que se hizo a posteriori fue la inclusión de **la gestión de errores de los archivos de entrada** de la aplicación. Al principio se había decidido aplicar un error estándar para todos los archivos de entrada que no cumplieran con el formato establecido (sección 3.2.1), pero dado que podrían existir series temporales con cientos de datos, se optó más adelante por dividirlos en dos tipos y decir en qué línea del archivo ocurrían. De esta forma no se perdería tiempo innecesariamente en buscar

dónde puede estar mal definida la entrada, ya que ningún experto desea usar su tiempo en arreglar esto.

3.2.3. Consideraciones matemáticas

Al realizar el análisis de las series temporales, es necesario hacer uso de fórmulas matemáticas para poder obtener la información de distintos símbolos. Por ejemplo, para saber la pendiente entre dos puntos, encontrar picos o hundimientos, calcular su duración y altura y hallar los pasos por cero.

Para el cálculo de la **pendiente entre dos puntos**, que nos permitirá conocer si nos encontramos ante una subida o bajada y si es aguda o suave (de acuerdo con un parámetro que le indiquemos), haremos uso de la Ecuación Punto-Pendiente, con la que, a partir de las coordenadas de dos puntos, se puede saber esta información. En la expresión 5 se muestra dicha ecuación.

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

Expresión 5. Ecuación Punto-Pendiente

Si la pendiente *m* es positiva, significa que nos encontramos ante un tramo de subida; si es negativa será una bajada; si es 0, será un tramo constante.

Una vez que sabemos cómo calcular la pendiente de cualquier tramo de la serie, podemos usarla para hallar pasos por cero, así como la duración de picos y hundimientos.

Para el caso de los **pasos por cero**, es necesario resolver un pequeños sistema de ecuaciones entre la recta *y=0* y la recta que une los dos puntos entre los que se pasa por cero, obteniéndose el valor de x en que se produce el corte. Para ello, como ya se ha comentado, se necesita saber primero la pendiente entre dichos puntos. Después, como se indica en las Expresiones 6-10, se resuelve el sistema y nos queda una expresión

reducida que nos permite calcular cualquier paso por cero sabiendo únicamente las coordenadas de dos puntos.

$$y = mx + b$$

Expresión 6. Ecuación de una recta

$$b = y_i - \frac{y_2 - y_1}{x_2 - x_1} x_i$$

Expresión 7. Cálculo de *b*

$$\begin{cases} \boldsymbol{y} = 0 \\ \boldsymbol{y} = \dfrac{y_2 - y_1}{x_2 - x_1} x + y_i - \dfrac{y_2 - y_1}{x_2 - x_1} x_i \end{cases}$$

Expresión 8. Sistema de ecuaciones a resolver

$$\frac{y_2 - y_1}{x_2 - x_1} x + y_i - \frac{y_2 - y_1}{x_2 - x_1} x_i = 0$$

Expresión 9. Sustitución de la variable '*y*' en la otra ecuación

$$\mathbf{x} = x_i - \frac{y_\mathrm{i}}{\left(\frac{y_2 - y_1}{x_2 - x_1}\right)}$$

Expresión 10. Ecuación para hallar la '*x*' del paso por cero (reducida)

De dichos conjuntos de expresiones, el significado de las variables escritas es el siguiente: ***m*** es la pendiente de la recta que corta al eje x (ya explicada en la Expresión 1), ***b*** es la intersección de dicha recta con el eje y, x_1 ***e*** y_1 son las coordenadas del primer punto de la recta de corte, x_2 ***e*** y_2 son las coordenadas del segundo punto de la recta de corte, x_i ***e*** y_i son las coordenadas de cualquiera de los dos puntos anteriores mencionados.

Así pues, con una sola ecuación (Expresión 10), podemos calcular cuáles son los puntos concretos en que se producen los cortes de la serie temporal con el eje x.

Para calcular la **duración de un pico o hundimiento**, que será un atributo a mostrar en la salida del sistema, es necesario tomar dos puntos de referencia. Tanto para picos como hundimientos se tomará como base el primer punto a su derecha o izquierda en que se produzca un cambio de tramo (es decir, de bajada a subida, de bajada a constante, de subida a bajada o de subida a constante). El otro punto de referencia será el que se encuentre al otro lado del pico o hundimiento, con la misma coordenada 'y' que el primero. En las figuras 10-13, se muestran cuatro ejemplos de la duración de picos y hundimientos.

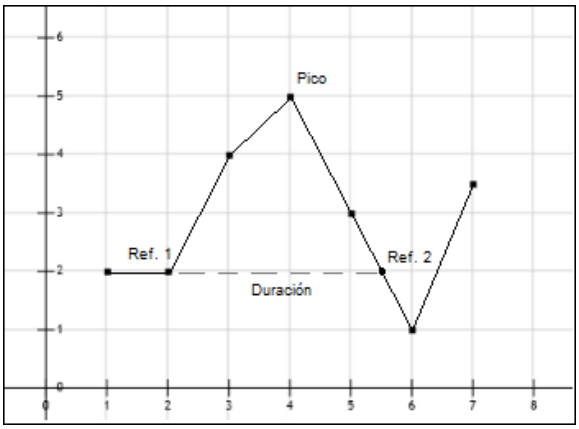


Figura 10. Ejemplo Pico 1

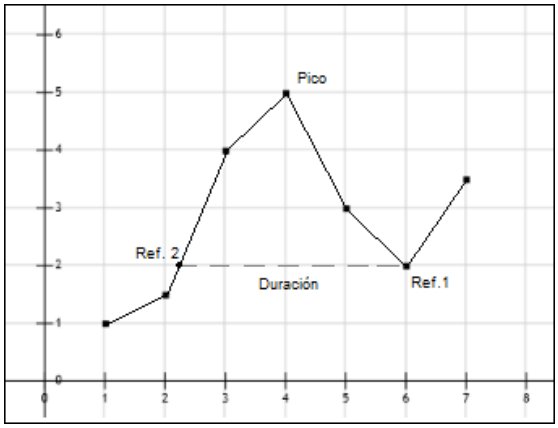


Figura 11. Ejemplo Pico 2

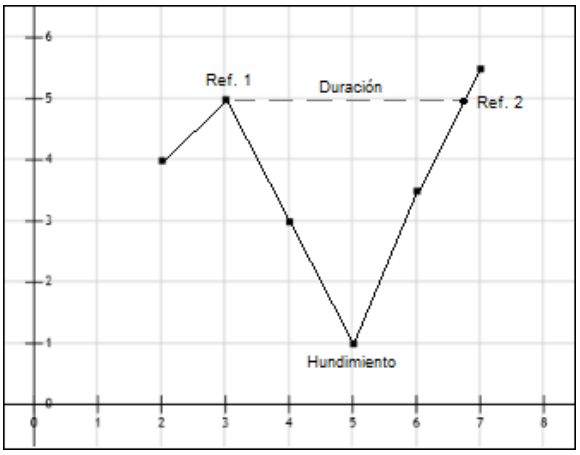


Figura 12. Ejemplo Hundimiento 1

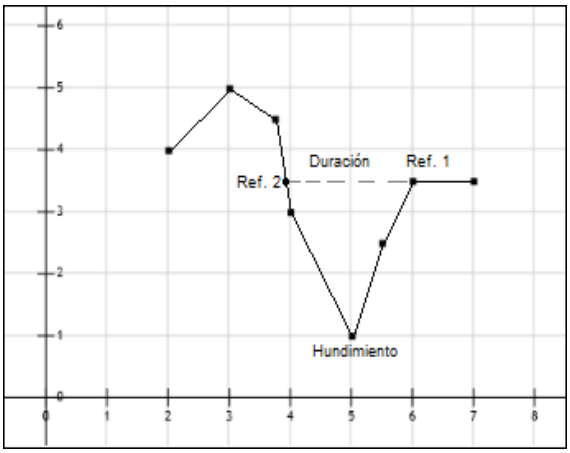


Figura 13. Ejemplo Hundimiento 2

Una vez explicado este término, pasamos a explicar cómo se calcula esta duración. En primer lugar, tomamos el primer punto de referencia, que necesariamente tendrá que estar definido en la serie temporal. Para hallar el siguiente punto habrá que resolver un sistema de ecuaciones entre la recta $y=c$ (siendo c la coordenada y del primer punto de referencia) y la recta que pasa por los puntos inmediatamente anterior y posterior a ese valor de 'y' al otro lado del pico o hundimiento. Por último, sólo habría que calcular el valor absoluto de la diferencia entre los dos puntos.

Se puede intuir que será un sistema muy parecido al calculado en el caso del paso por cero, sólo que en este caso se hará con la recta $y=c$ en lugar de $y=0$.

Las Expresiones 11-14 muestran la resolución de este sistema de ecuaciones, tomando como referencia las Expresiones 6 y 7, que son exactamente iguales para este caso.

$$\begin{cases} \boldsymbol{y} = c \\ \boldsymbol{y} = \dfrac{y_2 - y_1}{x_2 - x_1} x + y_i - \dfrac{y_2 - y_1}{x_2 - x_1} x_i \end{cases}$$

Expresión 11. Sistema de ecuaciones a resolver

$$\frac{y_2 - y_1}{x_2 - x_1} x + y_i - \frac{y_2 - y_1}{x_2 - x_1} x_i = \mathrm{c}$$

Expresión 12. Sustitución de la variable '*y*' en la otra ecuación

$$\mathrm{x} = x_i + \frac{\mathrm{c} - y_\mathrm{i}}{(\frac{y_2 - y_1}{x_2 - x_1})}$$

Expresión 13. Ecuación para hallar la '*x*' del punto de referencia (duración)

$$d = |x_{ref1} - x_{ref2}|$$

Expresión 14. Cálculo de la duración de un pico/hundimiento

Como se ve, lo único que cambia en este caso es que se añade el valor *c* a la ecuación final. El significado de las variables es el mismo que en el paso por cero, con el añadido de la susodicha constante c.

Una vez hallada la duración, se puede calcular la **altura del pico o hundimiento**, que es necesaria para saber si un pico o hundimiento será considerado como tal en la salida del sistema. Será el valor absoluto de la diferencia entre las coordenadas *'y'* del punto de pico o hundimiento y de cualquiera de los dos puntos anteriores hallados en la duración (x_{ref1}) ó (x_{ref2}). En la Expresión 15 se muestra este cálculo.

$$h = |x_{refi} - x_{pico/hund}|$$

Expresión 15. Cálculo de la altura de un pico/hundimiento

Como últimas consideraciones matemáticas, tratamos los **picos y hundimientos** y cómo se hallan. En circunstancias normales, en que se posea una ecuación para definir

una serie temporal, los picos y hundimientos serán todos aquellos puntos en que la derivada de la ecuación es 0. Sin embargo, dado que las series temporales para la aplicación serán definidas por medio de un conjunto de puntos, el cálculo de estos puntos límite se debe hacer de otra forma.

Para cada uno de los puntos, habrá que calcular si su coordenada *'y'* es mayor (picos) o menor (hundimientos) que los puntos inmediatamente anterior y posterior. En caso de ser así, añadiríamos estos puntos a la lista de puntos límite. De entre todos ellos, los que tengan la mayor y la menor coordenada *'y'* se les añadirá el atributo *absoluto*, el resto serán *locales*.

Hay un **caso concreto**, referente al concepto de Tiempo en Pico/Hundimiento, ya discutido en la sección anterior, en que podría haber picos y hundimientos, habiendo un tramo constante, si se cumple la condición de este parámetro de entrada. Igualmente, si la altura del pico o hundimiento no es mayor o igual que el parámetro de entrada Altura Mínima de Picos/Hundimientos, también discutido en la sección anterior, no será considerado en la salida.

Con todos estos cálculos matemáticos quedan definidos todos los símbolos que serán mostrados en la salida del sistema.

3.2.4. Dificultades encontradas

A lo largo de la implementación del módulo software, ha habido una serie de características que, dada su dificultad, han hecho que se le tenga que dedicar más tiempo del previsto inicialmente. En esta sección se van a tratar estas dificultades, explicando.

La primera de las dificultades fue la **familiarización con el entorno de trabajo** y con el lenguaje C#. Aunque es un lenguaje similar a los ya usados anteriormente, las dificultades fueron relativas a encontrar el entorno de trabajo correcto, ya que se probaron varios antes de decidirse por Visual Studio, y a familiarizarse con la forma en que se añaden bibliotecas, métodos, auto-compleción de campos, etc. [4]

Otra dificultad importante fue el llevar una **buena ordenación de todo el código** escrito y saber en qué métodos y clases se realizaban todas las acciones. Para ello, era necesario tener comentadas las líneas del código, de forma que no quedara ninguna variable ni método localizado. Para hacer depuración esto también fue crucial, ya que sin comentarios, muchas veces no habría constancia de dónde se producía el error.

Uno de las mayores dificultades encontradas fue la del **cálculo de la duración** de un pico o hundimiento. En un principio, al buscar información sobre cómo se hacía el cálculo de este atributo, se encontraron distintas formas de considerarla: en algunos casos se definía como la distancia entre los puntos inmediatamente anterior y posterior al punto límite, y en otros como la distancia entre los puntos de cambio de tramo anterior y posterior al punto límite. Finalmente se encontró la definición correcta de *duración*, pero para entonces ya se habían implementado los anteriores cálculos en el sistema y había conllevado bastante pérdida de tiempo. Igualmente, la forma correcta de considerar la duración fue bastante complicada de implementar, ya que, como se ha explicado en la sección 3.2.3, hay que hacer una interpolación entre dos puntos que pueden estar muy alejados de cualquier referencia, lo que hace que se puedan hacer los cálculos entre puntos erróneos.

Otra dificultad más fue el problema que acarreaba trabajar en un equipo español, que **separa los números decimales con una coma en lugar del punto**. Por ello, en la salida del sistema los números se mostraban separados por comas y, como el archivo de salida generado era de extensión .csv, todas las comas que encontraba las consideraba separadores, con el consiguiente descuadre de información. Solucionarlo fue bastante complicado, ya que hasta que se encontró la solución satisfactoria, las anteriores no sólo cambiaban las comas de los resultados, sino que alteraban las de todos los lugares del sistema en que aparecían.

CAPÍTULO 5: RESULTADOS

En esta sección se van a discutir los resultados que produce el sistema, una vez que ya se han tratado todos los temas de diseño y consideraciones de la propia aplicación.

Como ya se ha comentado, en la salida se mostrarán los símbolos que se corresponden con los eventos que suceden en la serie temporal de entrada, que tienen asociado un atributo a ellos y se mostrarán en tres columnas diferentes datos asociados a ellos.

Estos símbolos podrán variar en la salida dependiendo de los parámetros de entrada que configure el usuario: Pendiente límite para subidas/bajadas, Altura mínima de picos/hundimientos y Tiempo en Límite para picos/hundimientos (sección 3.2.2).

La información que se muestra en la salida no permite reconstruir la serie temporal exactamente como está definida en la entrada, ya que lo que se pretende no es mostrar la misma información que ya se tiene en la entrada, sino los comportamientos que presenta la serie temporal por medio de símbolos y atributos, que permiten clasificar mejor la información y hacer análisis y comparaciones de forma más rápida y eficiente. Ahora bien, sí se mostrará otra información relevante, como el instante de tiempo en que ocurren los distintos eventos, que sí puede ser necesaria en análisis posteriores.

En las figuras 14 y 15, se muestra la salida como se puede visualizar en la propia aplicación creada y en el archivo csv generado aparte.

Subida.aguda ,	1	,	3	,	3.5
Pico.local ,	1.571429	,	4	,	2.5
Bajada.aguda ,	3	,	4	,	2.5
Hundimiento.local ,	3	,	4.3125	,	2.5
Subida.aguda ,	4	,	5	,	8
Pico.absoluto ,	4	,	6.166667	,	8
Bajada.aguda ,	5	,	7	,	13
PasoPorCero ,	6.5	,	-	,	-
Constante ,	7	,	8	,	-3
Bajada.suave ,	8	,	9.5	,	1
Hundimiento.absoluto ,	8	,	9.666667	,	1
Subida.aguda ,	9.5	,	10	,	3
Pico.local ,	9.833333	,	11	,	1
Bajada.suave ,	10	,	11	,	1
Constante ,	11	,	12	,	-2
Subida.aguda ,	12	,	13	,	9
PasoPorCero ,	12.22222	,	-	,	-

Figura 14. Ejemplo de la salida del sistema en la aplicación

	A	B	C	D
1	Subida.aguda	1	3	3.5
2	Pico.local	1.571429	4	2.5
3	Bajada.aguda	3	4	2.5
4	Hundimiento.local	3	4.3125	2.5
5	Subida.aguda	4	5	8
6	Pico.absoluto	4	6.166667	8
7	Bajada.aguda	5	7	13
8	PasoPorCero	6.5	-	-
9	Constante	7	8	-3
10	Bajada.suave	8	9.5	1
11	Hundimiento.absoluto	8	9.666667	1
12	Subida.aguda	9.5	10	3
13	Pico.local	9.833333	11	1
14	Bajada.suave	10	11	1
15	Constante	11	12	-2
16	Subida.aguda	12	13	9
17	PasoPorCero	12.22222	-	-

Figura 15. Ejemplo de la salida del sistema en el archivo csv

La información que se muestra en la salida es la siguiente:

- *Subida/Bajada*: Cuando un tramo de la serie temporal suba o baje, se mostrará este símbolo. Puede tener los atributos 'suave' o 'aguda'; se mostrará uno u otro dependiendo de lo que se haya configurado en el parámetro ‘Pendiente Límite para subidas/bajadas’ (consultar sección 3.2.2).

- *Pico/Hundimiento*: Cuando la coordenada 'y' de un punto esté por encima (Pico) o por debajo (Hundimiento) de la de su punto anterior y posterior y, además, se cumplan las condiciones de altura mínima y tiempo en hundimiento (consultar sección 3.2.2), se mostrará este símbolo. Puede tener los atributos 'absoluto' (cuando la coordenada 'y' sea la menor -Hundimiento- o mayor -Pico- de la serie temporal) o 'local' (para el resto de casos).

- *Constante*: Cuando no varíe el valor de la coordenada 'y' de un tramo, al menos entre dos puntos, se mostrará este símbolo. No tiene atributos asociados.

- *PasoPorCero*: Cuando la serie temporal corte al eje 'x' de coordenadas, se mostrará este símbolo. No tiene atributos asociados.

En las siguientes tres columnas, B, C y D (Figura 15), se mostrará información diferente dependiendo de cada símbolo.

En la **primera columna** se muestra:

- Para los símbolos *'Subida', 'Bajada' y 'Constante'* el punto en el que empieza el tramo (coordenada 'x').
- Para los símbolos *'Pico' y 'Hundimiento'* el punto en el que empieza el punto límite (coordenada 'x'), es decir, desde dónde se comienza a contar su duración.
- Para el símbolo *'PasoPorCero'* el punto en el que corta al eje 'x'.

En la **segunda columna** se muestra:

- Para los símbolos *'Subida', 'Bajada' y 'Constante'* el punto en el que termina el tramo (coordenada 'x').
- Para los símbolos *'Pico' y 'Hundimiento'* el punto en el que termina el punto límite (coordenada 'x'), es decir, hasta donde se cuenta su duración.
- Para el símbolo *'PasoPorCero*' no se muestra más información.

En la **tercera columna** se muestra:

- Para los símbolos *'Subida' y 'Bajada'* la altura desde el punto de inicio del tramo hasta el punto de fin (diferencia entre las coordenadas 'y' de ambos puntos).
- Para los símbolos *'Pico' y 'Hundimiento'* la altura de dicho punto límite, es decir, desde cualquiera de los puntos de inicio o fin hasta el punto límite en que ocurre el pico o hundimiento (diferencia entre las coordenadas 'y' de ambos puntos).
- Para el símbolo *'Constante'* el valor de la coordenada 'y' que se mantiene constante mientras dura este tramo.
- Para el símbolo *'PasoPorCero'* no se muestra más información.

Para que el archivo csv que se generará en la salida pueda ser visualizado en columnas correctamente, es posible que se necesite cambiar la configuración de separadores de los programas lectores de csv que se usan en cada equipo que lo ejecute. En la ayuda de la aplicación se incluye una solución para programas de Microsoft.

Para comprobar que toda la información mostrada es lo suficientemente útil, se han validado los resultados en dos dominios diferentes.

4.1. Validación en el campo de la Medicina

Como se explica en [Aguirre et al. 94], el análisis de series temporales en el campo de las Ciencias de la Salud es muy importante para diversas tareas, como son las predicciones, el control y simulación de procesos o la generación de nuevas teorías biológicas.

Así pues, para el campo de la Medicina, se evaluará la información de una gráfica que contiene los valores de nuevos casos de diabetes que se han diagnosticado en los últimos 30 años en adultos de EE.UU [5]:

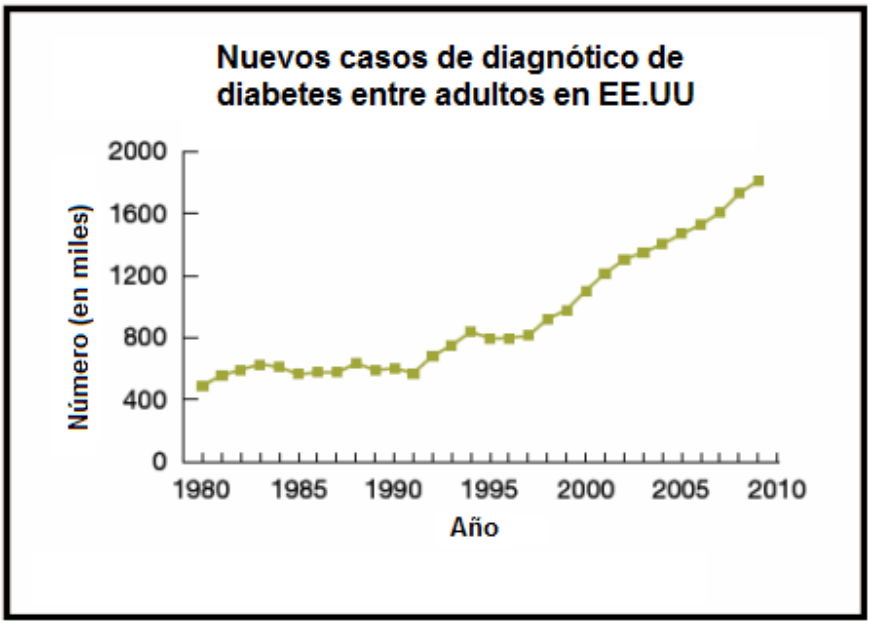


Figura 16. Nuevos casos de diabetes en adultos en EE.UU.

Al transformar los valores de esta serie a través de la aplicación, obtenemos la siguiente salida:

	A	B	C	D
1	Subida.aguda	1980	1982	160
2	Subida.suave	1982	1983	40
3	Bajada.aguda	1983	1985	130
4	Subida.suave	1985	1987	20
5	Subida.aguda	1987	1988	80
6	Bajada.aguda	1988	1989	70
7	Subida.suave	1989	1990	10
8	Bajada.suave	1990	1991	25
9	Subida.aguda	1991	1992	185
10	Subida.suave	1992	1993	50
11	Subida.aguda	1993	1994	70
12	Bajada.suave	1994	1995	40
13	Subida.suave	1995	1999	100
14	Subida.aguda	1999	2003	400
15	Subida.suave	2003	2005	60
16	Subida.aguda	2005	2006	110
17	Subida.suave	2006	2008	90
18	Subida.aguda	2008	2010	160

Figura 17. Resultados de la serie temporal médica

Ya que los valores de tiempo avanzan de uno en uno, pero los del otro eje avanzan en grandes cantidades, ha sido necesario ajustar los parámetros de Pendiente Límite y Altura Mínima para que no se mostraran siempre atributos agudos o picos/hundimientos donde en realidad no interesaba mostrarse. Pendiente límite ha sido establecido en 50 y Altura Mínima en 150. El parámetro de Tiempo en Límite sigue en el valor predeterminado, que es 0.

Podemos observar que los únicos símbolos que se muestran son las subidas y bajadas, ya que los picos y hundimientos, a no ser que fueran elementos muy diferenciados, no son importantes para los resultados.

Estos resultados nos indican, además, los años entre los que se produce una subida o bajada, tanto aguda o suave, y la diferencia de casos que se producen entre su inicio y fin; es una información condensada perfectamente para lo que un experto en la materia desea saber.

De esta forma, además, puede hacer comparaciones con otras bases de datos de series temporales de nuevos casos de otras enfermedades y poder contrastar información de forma mucho más clara y rápida.

4.2. Validación en el campo de la Ingeniería

Como se detalla en [Chatfield et al. 03], el campo de la Ingeniería también precisa de análisis de series temporales, siendo muy importante en tareas similares a las discutidas en el campo de la Medicina, como el desarrollo de nuevos modelos y su comparación con aproximaciones previas en términos de la precisión del pronóstico.

Para este campo de la Ingeniería se va a analizar una serie en la que se han obtenido la temperatura media anual en Italia durante 60 años [6]:

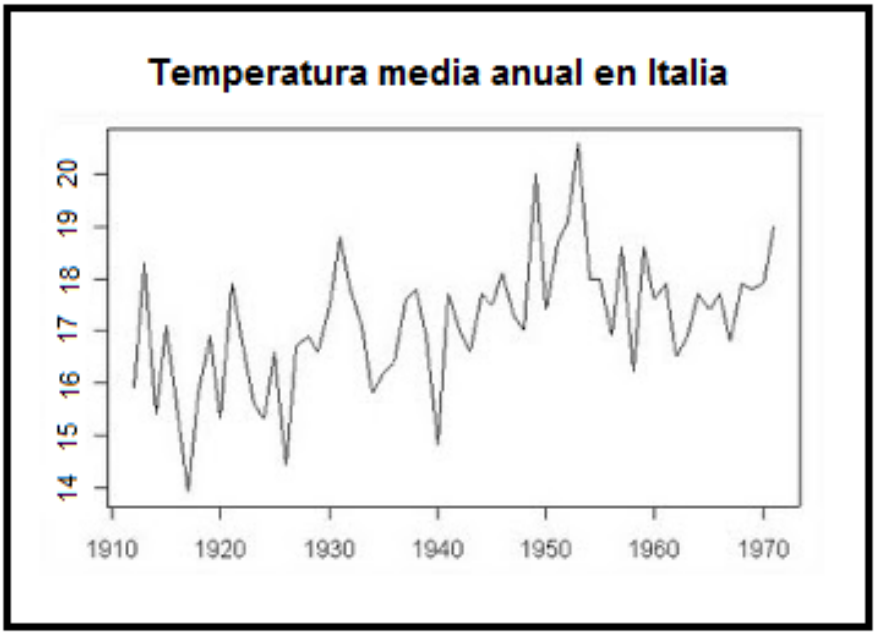


Figura 18. Temperatura media anual en Italia

Al transformar los valores de esta serie a través de la aplicación, obtenemos la siguiente salida:

	A	B	C	D
1	Subida.aguda	1913	1914	2
2	Pico.local	1913	1914.8	2
3	Bajada.aguda	1914	1915	2.5
4	Hundimiento.local	1914.4	1916	1.5
5	Subida.suave	1915	1916	1.5
6	Pico.local	1915	1916.5	1.5
7	Bajada.aguda	1916	1917	3
8	Hundimiento.absoluto	1916	1919	3
9	Subida.aguda	1917	1918	2
10	Subida.suave	1918	1919	1
11	Pico.local	1917.75	1920	1.5
12	Bajada.suave	1919	1920	1.5
13	Hundimiento.local	1919	1920.6	1.5
14	Subida.aguda	1920	1921	2.5
15	Pico.local	1920	1922	2.5
16	Bajada.aguda	1921	1922	2.5
17	Bajada.suave	1922	1923	0.5
18	Hundimiento.local	1921.6	1924	1.5
19	Subida.suave	1923	1924	1.5
20	Pico.local	1923	1924.75	1.5
21	Bajada.aguda	1924	1925	2
22	Hundimiento.local	1924	1926	2
23	Subida.aguda	1925	1926	2
24	Subida.suave	1926	1927	0.5
25	Bajada.suave	1927	1928	0.5
26	Subida.suave	1928	1930	2.5
27	Pico.local	1928	1932.5	2.5
28	Bajada.suave	1930	1933	3
29	Hundimiento.local	1930.667	1937	2
30	Subida.suave	1933	1937	2
31	Pico.local	1933	1938.5	2
32	Bajada.suave	1937	1938	1

	A	B	C	D
33	Bajada.aguda	1938	1939	2
34	Hundimiento.local	1937	1940	3
35	Subida.aguda	1939	1940	3
36	Pico.local	1939.5	1942	1.5
37	Bajada.suave	1940	1942	1.5
38	Subida.suave	1942	1943	1
39	Bajada.suave	1943	1944	0.2000008
40	Subida.suave	1944	1945	0.7000008
41	Bajada.suave	1945	1947	1.5
42	Hundimiento.local	1945	1947.429	1.5
43	Subida.aguda	1947	1948	3.5
44	Pico.local	1947.286	1949	2.5
45	Bajada.aguda	1948	1949	2.5
46	Hundimiento.local	1948	1951.667	2.5
47	Subida.suave	1949	1952	3
48	Pico.absoluto	1949.5	1953	2.5
49	Bajada.aguda	1952	1953	2.5
50	Constante	1953	1954	18
51	Bajada.suave	1954	1955	1
52	Subida.suave	1955	1956	1.5
53	Bajada.suave	1956	1957	1
54	Subida.suave	1957	1958	1
55	Bajada.suave	1958	1959	0.5
56	Constante	1959	1960	18
57	Bajada.suave	1960	1961	1.5
58	Subida.suave	1961	1963	1
59	Constante	1963	1965	17.5
60	Bajada.suave	1965	1966	0.5
61	Subida.suave	1966	1967	0.5
62	Constante	1967	1969	17.5
63	Subida.suave	1969	1970	1.5

Figuras 19 y 20. Resultados de la serie temporal de temperaturas

En este caso, los valores de tiempo de ambos ejes avanzan en escalas parecidas: de uno en uno para los valores temporales y de forma similar en el eje de temperaturas; por ello, en esta ocasión se ha decidido ajustar la Pendiente Límite en 1.5, ya que la variación de un grado entre año y año no es suficientemente aguda, y la Altura Mínima en 1.5, ya que no interesa estar obteniendo picos y hundimientos en todos los casos, pero sí en bastante de ellos. El Tiempo en Límite, al no tratarse en este caso, se deja en el valor predeterminado otra vez.

En esta ocasión se muestran tanto subidas y bajadas, como constantes y muchos picos y hundimientos, que es la información que en este caso se considera relevante. Se vuelve a obtener información de la diferencia de temperaturas entre tramos, los años en que se produjeron las subidas y bajadas y los valores estables.

Así, también pueden hacerse comparaciones con otras series temporales en que se incluyan temperaturas medias anuales de otros países o de regiones más concretas, de una forma más clara y precisa.

CAPÍTULO 5: CONCLUSIONES

Llegados al capítulo de las conclusiones, conviene echar la vista atrás y recordar cuáles eran los planteamientos que se propusieron inicialmente, pasando por el desarrollo del trabajo hasta llegar a los resultados.

Se pretendía crear un módulo software para un sistema de Minería de Datos que aplicara técnicas de Abstracción Temporal para transformar series numéricas temporales en secuencias simbólicas. La finalidad de este proceso era la reducción de la dimensionalidad de los datos (es decir, poder trabajar con un menor número de datos) y poder aplicar, más adelante, determinadas técnicas como la Programación Genética Dirigida por Gramáticas, para la creación de Modelos Simbólicos.

Por tanto, el módulo creado se encarga de transformar una serie temporal numérica en una secuencia simbólica que representa el contenido semántico de un dominio específico. Gracias a ello, se extrae de forma automática el conjunto e símbolos que se corresponden con los comportamientos de la serie que son relevantes en el dominio y que, por tanto, serían los que un experto querría identificar.

Para que el sistema diseñado fuera más fiel a la realidad, era necesario que la transformación de la serie se realizara de la misma forma que lo haría un experto en el dominio de aplicación de dicha serie. Así, se consideraron los símbolos que un experto utiliza habitualmente en los campos de medicina, ingeniería, fisiología, etc. De la misma forma, al usarse un lenguaje cercano a los usuarios que podrían valerse de la aplicación, los resultados son más comprensibles.

La creación de dicha aplicación pasó por varias fases en las que se decidió añadir nuevas funcionalidades a la vez que se tuvieron que reducir algunas otras, sobre todo las que respectaban al post-análisis de las series temporales.

Por un lado, se añadió el símbolo de Paso por Cero a la lista propuesta inicialmente, ya que se consideró información suficientemente relevante para varios dominios. Igualmente, se incluyeron parámetros que podría configurar el usuario y se amplió el número de atributos que se mostrarían en la salida del sistema; de esta forma, quedó

unificado el formato de salida para que todos los símbolos y sus atributos aparecieran en un lado y el resto de información bien diferenciada en el otro.

Por otro lado, debido a la falta de tiempo, la fase de aplicación de un algoritmo de Minería de Datos que permitiera probar lo anteriormente diseñado, se tuvo que apartar y dejarlo como una futura línea de continuación del trabajo.

Tras crear la aplicación de transformación de símbolos, se hicieron pruebas en varios dominios que permitieron validar el sistema, al obtenerse la información correctamente para cada uno de ellos.

La transformación de una serie al dominio simbólico abre numerosas posibilidades de aplicación en varios dominios dentro de los ámbitos ya mencionados: caracterización funcional por distintos atributos, clasificación por ámbitos, detección de puntos conflictivos y errores en la captura de información inicial, etc. En el siguiente capítulo se indican todas las áreas en las que se puede avanzar, ya sea investigando en nuevos ámbitos o mejorando el módulo software.

CAPÍTULO 6: FUTURAS LÍNEAS DE TRABAJO

Tras la realización de este trabajo, son muchas las líneas que quedan abiertas para seguir profundizando en el tema de análisis de series temporales, incluyendo la mejora del sistema creado. De hecho, el punto en el que se deja el trabajo es donde más líneas de trabajo se abren en las que sería interesante avanzar. Algunas ya han sido tratadas en otros trabajos de investigación, como se comenta en anteriores capítulos, pero otras también merecen ser mencionadas:

- **Realizar medidas de la similitud** entre dos secuencias simbólicas como punto de partida para la tarea de comparación exhaustiva entre ellas. De esta forma, en campos como la medicina o la ingeniería, será más fácil distinguir los puntos de interés en eventos similares, focalizando en su información semántica.

- **Crear modelos de referencia** para análisis posteriores de las series temporales, entendiendo como modelo de un conjunto de secuencias simbólicas, una secuencia simbólica representativa de todo el grupo.

- **Aumentar el conjunto de símbolos detectados por el sistema.** Dependiendo del ámbito en el que se use el sistema, podría ser deseable mostrar más información de la que ya se proporciona en la salida. Por ejemplo, los puntos de inflexión (donde la segunda derivada es cero) o la distancia que hay entre dos eventos concretos de la serie también podrían ser interesantes de representar.

- **Adjuntar el dibujo de la serie temporal analizada** junto con la salida de los símbolos del sistema. Así, sería mucho más fácil comprobar las zonas en que ocurren eventos parecidos y la forma que tiene cada uno de los símbolos analizados.

- **Permitir que el usuario final pueda decidir qué símbolos** aparecerán en la salida del sistema. Dependiendo de la información en la que se quiera centrar, se prescindiría de toda la información accesoria y sólo se mostraría lo importante.

Esto sería más útil en series temporales que contuvieran muchos símbolos y fuera difícil filtrar la información relevante de la salida.

- **Incorporar un método de aprendizaje** en el sistema usado de transformación a símbolos con el objetivo de que los valores que toman los parámetros de estos algoritmos puedan ser modificados automáticamente según el experto vaya clasificando los resultados de los métodos. Esta inclusión ayudaría a perfeccionar los resultados obtenidos de ambos métodos, al estar automáticamente aprendiendo de las entradas suministradas por el experto.

- **Asociar valores de certeza borrosos** a los símbolos obtenidos por la aplicación para permitir un grado de flexibilidad en la definición de los símbolos resultantes de la traducción de una serie temporal. Esto permitiría que, por ejemplo, las subidas no tengan que ser siempre 100% suaves o agudas. En este caso habría que adaptar el método de creación de modelos simbólicos para que se tuviera en cuenta esta cuestión.

- **Diseñar un método de descubrimiento de patrones basado en secuencias simbólicas.** Este algoritmo permitirá identificar patrones simbólicos, es decir secuencias simbólicas concretas susceptibles de representar un determinado grupo de población o circunstancia especial, dependiendo del campo el que se aplicara este diseño.

CAPÍTULO 7: BIBLIOGRAFÍA

Libros de investigación:

[Aguirre et al. 94] Aguirre Jaime, A., *Introducción al tratamiento de series temporales. Aplicación a las Ciencias de la Salud*, Ed. Díaz de Santos, Madrid 1994.

[Chatfield et al. 03], Chatfield, C., *The Analysis of Time Series: An Introduction*, Ed. Chapman and Hall, Londres 2003.

[Sang-Wook et al. 06] Sang-Wook, K., Jeehee, Y., Sanghyun, P., Jung-Im, W., "Shape-based retrieval in time-series databases", *Journal of Systems and Software*, Volume 79, Issue 2, pp. 191–203, Febrero 2006.

Tesis doctorales:

[Santamaría et al. 11], A. Santamaría, "Modelo de Descubrimiento de Conocimiento para Series Temporales Numéricas aplicando Métodos" M.S. Thesis, DLSIIS, Universidad Politécnica de Madrid, 2011.

Estándares:

[1] IEEE *Criteria for Software Requirements*, IEEE Standard 830, 1998.

[2] ISO/IEC *Information Security Management System*, ISO/IEC 27001, 2005.

Fuentes online:

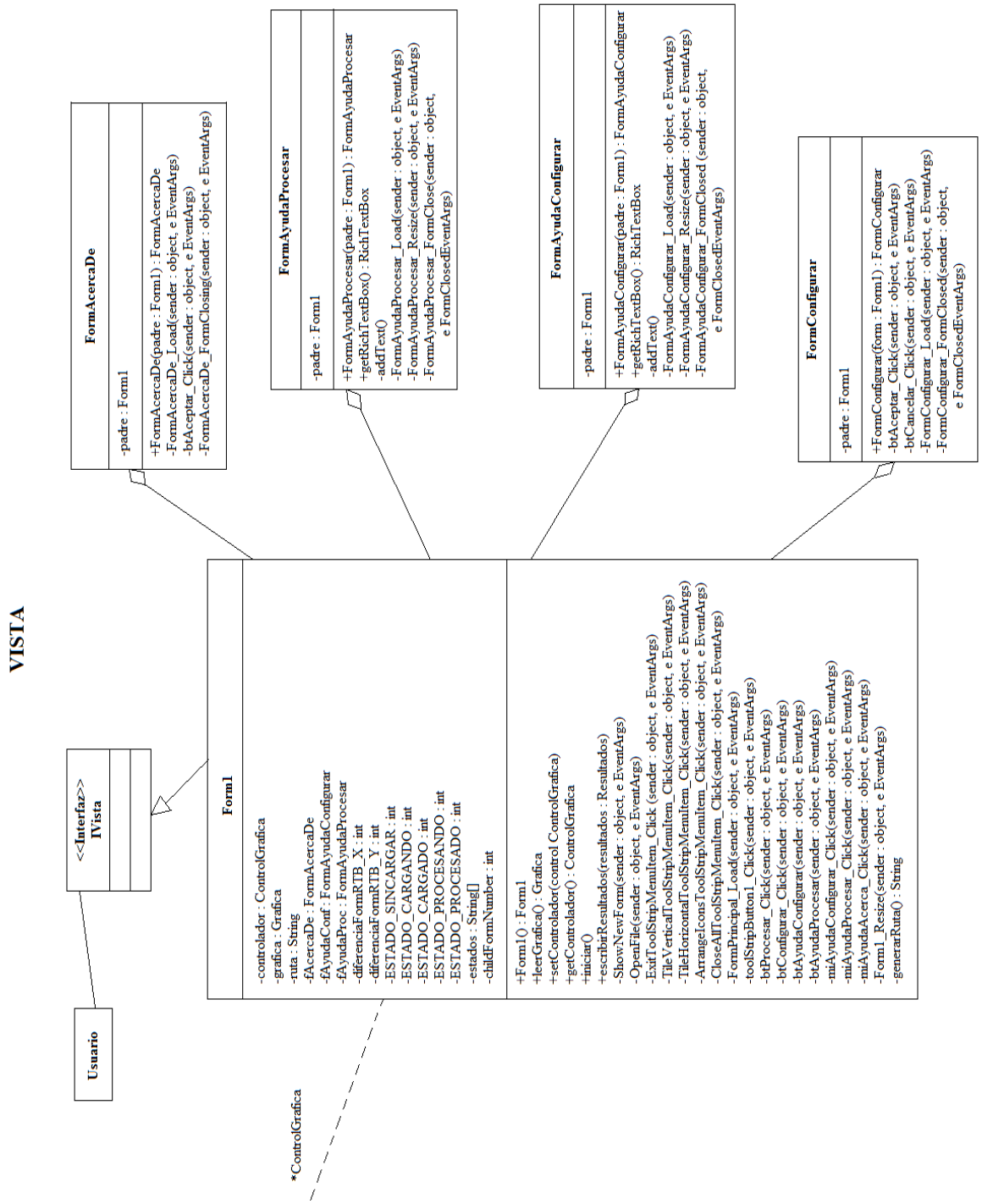
[3] Carlos Marín (2009). *El Proceso de KDD.* [Online]. Available: www.mineriadatos.blogspot.com.es/2009/04/el-proceso-de-kdd.html

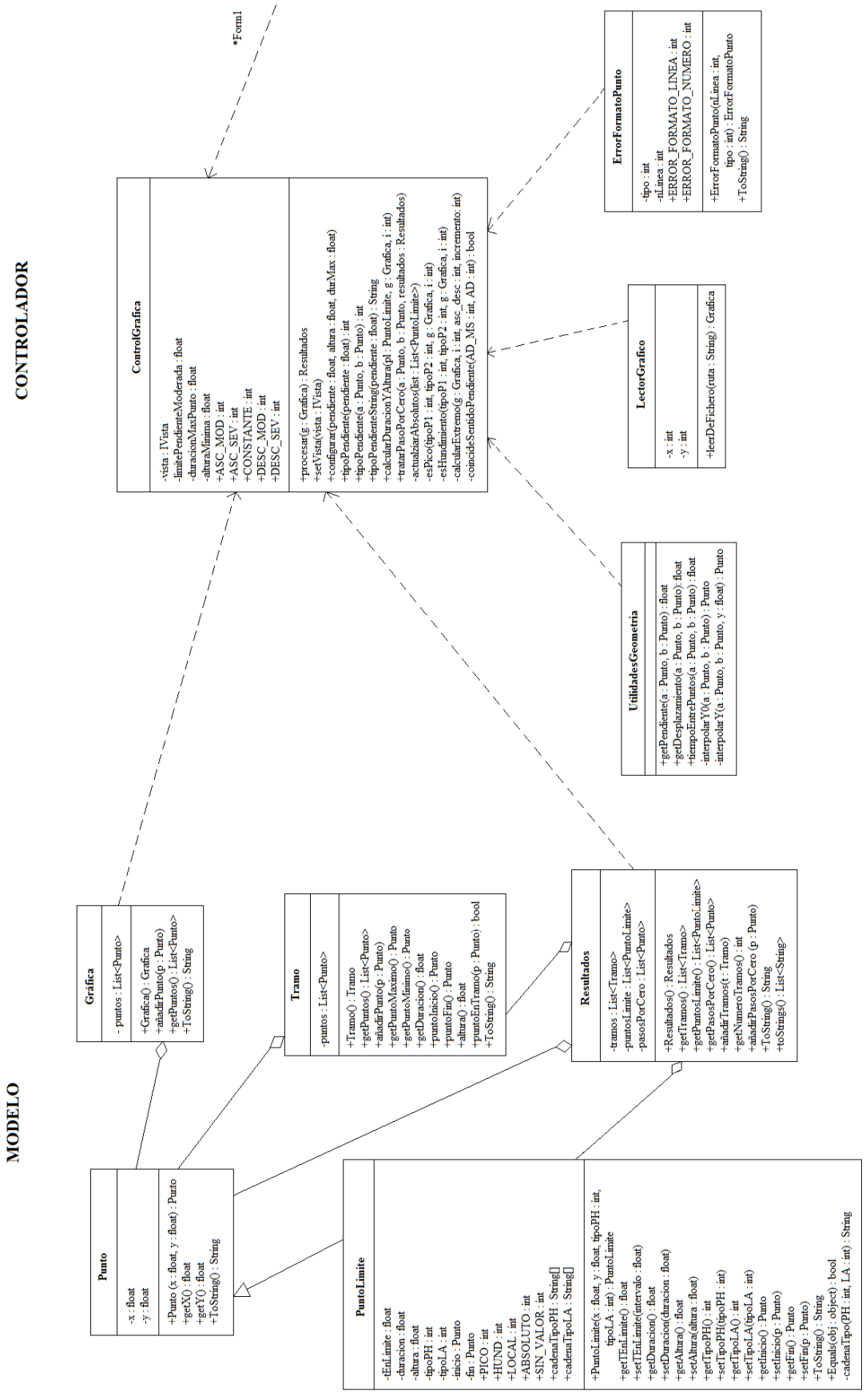
[4] Microsoft. (2012). *Visual Studio C# Libraries and References.* [Online]. Available: www.msdn.microsoft.com/es-ES/library/vstudio/

[5] Division of Diabetes Translation. (2011). *National Diabetes Fact Sheet.* [Online]. Available: www.cdc.gov/diabetes/pubs/pdf/ndfs_2011.pdf

[6] Ambiente e Territorio (2009). *Andamento meteo-climatico in Italia*. [Online]. Available:http://www3.istat.it/salastampa/comunicati/non_calendario/20100401_00/testointegrale20100401.pdf

ANEXO: DIAGRAMA DE CLASES




CONTROLADOR
MODELO
*Form1
ControlGrafica
-vista : IVista
-limitePendienteModerada : float
-duracionMaxPunto : float
-alturaMinima : float
+ASC_MOD : int
+ASC_SEV : int
+CONSTANTE : int
+DESC_MOD : int
+DESC_SEV : int
+procesar(g : Grafica) : Resultados
+setVista(vista : IVista)
+configurar(pendiente : float, altura : float, durMax : float)
+tipoPendiente(pendiente : float) : int
+tipoPendiente(a : Punto, b : Punto) : int
+tipoPendienteString(pendiente : float) : String
+calcularDuracionYAltura(pl : PuntoLimite, g : Grafica, i : int)
+tratarPasoPorCero(a : Punto, b : Punto, resultados : Resultados)
-actualizarAbsolutos(list : List<PuntoLimite>)
-esPico(tipoP1 : int, tipoP2 : int, g : Grafica, i : int)
-esHundimiento(tipoP1 : int, tipoP2 : int, g : Grafica, i : int)
-calcularExtremo(g : Grafica, i : int, asc_desc : int, incremento: int)
-coincideSentidoPendiente(AD_MS : int, AD : int) : bool
ErrorFormatoPunto
-tipo : int
-nLinea : int
+ERROR_FORMATO_LINEA : int
+ERROR_FORMATO_NUMERO : int
+ErrorFormatoPunto(nLinea : int, tipo : int) : ErrorFormatoPunto
+ToString() : String
LectorGrafico
-x : int
-y : int
+leerDeFichero(ruta : String) : Grafica
UtilidadesGeometria
+getPendiente(a : Punto, b : Punto) : float
+getDesplazamiento(a : Punto, b : Punto): float
+tiempoEntrePuntos(a : Punto, b : Punto) : float
-interpolarY0(a : Punto, b : Punto) : Punto
-interpolarY(a : Punto, b : Punto, y : float) : Punto
Grafica
- puntos : List<Punto>
+Grafica() : Grafica
+añadirPunto(p : Punto)
+getPuntos() : List<Punto>
+ToString() : String
Tramo
-puntos : List<Punto>
+Tramo() : Tramo
+getPuntos() : List<Punto>
+añadirPunto(p : Punto)
+getPuntoMaximo() : Punto
+getPuntoMinimo() : Punto
+getDuracion() : float
+puntoInicio() : Punto
+puntoFin() : Punto
+altura() : float
+puntoEnTramo(p : Punto) : bool
+ToString() : String
Resultados
-tramos : List<Tramo>
-puntosLimite : List<PuntoLimite>
-pasosPorCero : List<Punto>
+Resultados() : Resultados
+getTramos() : List<Tramo>
+getPuntosLimite() : List<PuntoLimite>
+getPasosPorCero() : List<Punto>
+añadirTramos(t : Tramo)
+getNumeroTramos() : int
+añadirPasosPorCero (p : Punto)
+ToString() : String
+toStrings() : List<String>
Punto
-x : float
-y : float
+Punto (x : float, y : float) : Punto
+getX() : float
+getY() : float
+ToString() : String
PuntoLimite
-EnLimite : float
-duracion : float
-altura : float
-tipoPH : int
-tipoLA : int
-inicio : Punto
-fin : Punto
+PICO : int
+HUND : int
+LOCAL : int
+ABSOLUTO : int
+SIN_VALOR : int
+cadenaTipoPH : String[]
+cadenaTipoLA : String[]
+PuntoLimite(x : float, y : float, tipoPH : int, tipoLA : int) : PuntoLimite
+getTEnLimite() : float
+setTEnLimite(intervalo : float)
+getDuracion() : float
+setDuracion(duracion : float)
+getAltura() : float
+setAltura(altura : float)
+getTipoPH() : int
+setTipoPH(tipoPH : int)
+getTipoLA() : int
+setTipoLA(tipoLA : int)
+getInicio() : Punto
+setInicio(p : Punto)
+getFin() : Punto
+setFin(p : Punto)
+ToString() : String
+Equals(obj : object) : bool
-cadenaTipo(PH : int, LA : int) : String

Este documento esta firmado por

	**Firmante**	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
	Fecha/Hora	Fri Feb 14 19:49:40 CET 2014
	Emisor del Certificado	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
	Numero de Serie	630
	Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)